# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

| | | |
|---|---|---|
| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE<br>July 1994 | 3. REPORT TYPE AND DATES COVERED<br>Final 8 Feb 91-16 May 94 |

| | |
|---|---|
| 4. TITLE AND SUBTITLE<br>Parallelism Detection and Scheduling Strategies for Reliable and Efficient Execution on Multicomputers | 5. FUNDING NUMBERS<br>DAAL03-91-G-0031 |
| 6. AUTHOR(S)<br>Dharma P. Agrawal | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>North Carolina State University<br>Raleigh, NC 27607 | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)<br>U.S. Army Research Office<br>P.O. Box 12211<br>Research Triangle Park, NC 27709-2211 | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER<br>ARO 28192.22-MA |

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

| | |
|---|---|
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT<br>Approved for public release; distribution unlimited. | 12b. DISTRIBUTION CODE |

13. ABSTRACT *(Maximum 200 words)*

Introduced a new concurrent object-oriented language called CORE which solves the inheritance anomaly problem.

By separating and localizing the synchronization schems from the main bodies of the program components, CORE allows a high degree of encapsulation and re-use for the synchronization code and the program modules.

19941129 005

(cont'd on reverse side)

| | |
|---|---|
| 14. SUBJECT TERMS<br>Parallelism Detection, Scheduling Strategies, Multicomputers, Distributed Memory Multiprocessors | 15. NUMBER OF PAGES |
| | 16. PRICE CODE |

| | | | |
|---|---|---|---|
| 17. SECURITY CLASSIFICATION OF REPORT<br>UNCLASSIFIED | 18. SECURITY CLASSIFICATION OF THIS PAGE<br>UNCLASSIFIED | 19. SECURITY CLASSIFICATION OF ABSTRACT<br>UNCLASSIFIED | 20. LIMITATION OF ABSTRACT<br>UL |

# Parallelism Detection & Scheduling Strategies For Reliable & Efficient Execution on Multicomputers

FINAL REPORT

Author: Dharma P. Agrawal

July 31 1994

U.S. ARMY RESEARCH OFFICE

Contract Number: DAAL03-91-G-0031

North Carolina State University
Raleigh, NC 27695-7911.
Ph: (919)-515-3984
Fax: (919)-515-5523

# Brief Research Outline

The popularity of Distributed Memory Multiprocessors (DMM) has been on the rise in recent years. The increase can be attributed to the advances in VLSI technology, better inter-processor communication networks and more efficient routing algorithms. Some of the applications using DMMs are fluid flow, weather modeling, database systems, etc. The object-oriented design methodology is being widely used to solve these practical problems.

To obtain the maximum benefits of DMMs, an efficient task partitioning and scheduling strategy is essential. The task partitioning algorithm partitions the application into separate tasks by detecting the parallelism in the program and represents them in the form of a Directed Acyclic Graph (DAG). After the application is transformed to a DAG, the tasks are scheduled onto the processors. This work deals with automating the process of parallelism detection to obtain better partitions of the applications and then scheduling the partitioned tasks to achieve reduced overall execution time.

As the title of the project suggests, there are two main objectives of this research. Those two objectives are described below in detail.

## 1 Parallelism Detection

There are several object-oriented languages but they fail to integrate concurrency and object-orientedness. Even the available concurrent object-oriented languages suffer from *inheritance anomaly*, which forces non-trivial class re-definitions because the synchronization code cannot be effectively inherited. This research has led to the development of a new concurrent object-oriented language, *CORE*, which does not suffer from inheritance anomaly and allows a programmer to specify parallel tasks at the inter- and intra-object levels. To achieve the best results of running any program on a parallel machine, the compiler needs to automatically detect the inherent parallelism in the program. The current goals of this research are to investigate the techniques for automatically detecting parallelism in object-oriented programs at the compile time.

A concurrent object-oriented language CORE has been developed which minimizes inheritance anomaly, promotes code reuse of both the sequential and parallel components, and offers mechanisms for controlling computational granularity. The methodology presented in this research is extendible to any class based language.

## 2 Scheduling

Efficient use of resources requires proper scheduling algorithm. There can be several goals of any scheduling algorithm. This research considers reducing the overall execution time of any program as the primary goal. A *Threshold Scheduling* scheme has been proposed for DMMs. In this scheduling scheme the compile time schedule is found using a new concept of *threshold* of a task that quantifies a trade-off between the schedule length and the degree of parallelism. In case of multiple tasks competing for the same processor, the tie is broken using a merit function of the tasks on the processor that examines a match between a task and a processor. The threshold is varied globally, and the best value to satisfy one of the following scheduling goals is found:

- Compiling for minimizing schedule length: Suitable for large systems.
- Compiling for processor requirements below a given maximum number of available processors in the system: Suitable for small systems.

A run-time support for Sisal compiler has been developed for Intel Gamma. Delta and Paragon machines. The run-time support helped correlate the compile time estimated results versus the actual run-time results. The run-time results for Sisal compiler using threshold scheduling scheme were comparable to the compile-time-estimated results. The speedups obtained on all three machines were lower than the predicted values. This is primarily due to the run-time overheads and also due to the assumptions made in modeling communications. Of the three machines, i.e. Intel Delta, Gamma and Paragon, Paragon was closest to the predicted values followed by Delta and Gamma. Paragon was better than the other two because of better hardware and also because it allows multiprogramming. The results obtained are promising and this scheme could be used for compiling programs for varying sizes of DMMs.

Another scheduling algorithm for DMMs, namely Search and Duplication Based Scheduling (SDBS) algorithm, has been developed. This algorithm is based on certain realistic assumptions and if they are satisfied, the algorithm generates an optimal time schedule. If the conditions are satisfied then no lower execution time schedule can be generated, and this has been proved. Even if the assumptions are not satisfied, the SDBS algorithm generates a schedule which is close to optimal. The results for such a case have been obtained with the help of extensive simulations. SDBS algorithm has been enhanced to generate optimal schedule in cases where the task execution times are varying. A task encapsulates control dependencies within it. Thus for tasks which encapsulate loops or if-then-else type of statements, the execution time could vary. The enhanced SDBS algorithm takes such tasks into consideration and produces an optimal schedule.

The schedule time obtained by using SDBS algorithm for random DAGs was compared against the level of the entry node. The level of the entry node is the lowest possible schedule time because the communication times are neglected when calculating levels. The schedule time obtained by SDBS algorithm was 11% over the absolute lowerbound on the average. The lowerbound cannot be achieved in most of the cases because of the communication time. The performance of enhanced SDBS algorithm was compared against the SDBS algorithm for cases where the task execution times were varying. For a small DAG the enhanced SDBS algorithm is better than SDBS algorithm by 4% on the average. This figure is low because this comparison was performed on a small DAG with only a few tasks having varying execution times. For larger DAGs with more number of varying execution time tasks we would expect the enhanced SDBS algorithm to perform much better then SDBS algorithm.

## 3 Significant Research Results

- Introduced a new concurrent object-oriented language called CORE which solves the inheritance anomaly problem.
- By separating and localizing the synchronization schems from the main bodies of the program components, CORE allows a high degree of encapsulation and re-use for the synchronization code and the program modules.
- Introduced a new compile time method to schedule functional parallelism in a program on distributed memory machines.
- Scheduling goals for the scheduling method can be either to reduce execution time or to generate a schedule to meet the system size.
- Concept of *threshold* introduced to quantify the trade-off between the completion time and the degree of parallelism in the schedule.

- Scheduler incorporated in the compiler backend for targeting Sisal to Intel Touchstone i860 systems.
- Introduced a new scheduling scheme. SDBS for scheduling tasks represented in the form of a DAG onto Distributed Memory Machines.
- SDBS algorithm is optimal and if assumptions are satisfied a lower execution time schedule cannot be generated.
- Enhanced SDBS algorithm developed which takes tasks with varying execution times into account.

## 4 List of Publications

1. J.C. Conrad and D.P. Agrawal. "Asynchronous Parallel Arc Consistency Algorithms on a Distributed Memory Machine," *Journal of Parallel and Distributed Computing*, 1994. to appear.
2. T.Y. Chung, N.K. Sharma, and D.P. Agrawal. "Cost-performance tradeoffs in Manhattan Street Networks vs 2-D Torus," *IEEE Transactions on Computers* , vol. 43. no. 2. February 1994. pp. 240-243.
3. S.S. Pande. D.P. Agrawal. and J. Mauney. "SISAL on Distributed Memory Machines, Proceedings of Third Sisal Conference. San Diego. Oct. 3-5. 1993.
4. S.S. Pande. D.P. Agrawal, and J. Mauney, "Automatic Compiler for a Parallel Functional Language on a Distributed Memory Machine", IEEE Parallel and Distributed Technology, vol. 2. no. 1. Spring 1994, pp. 64-76.
5. S.S. Pande. D.P. Agrawal. and J. Mauney, "A Scalable Scheduling Method for Functional Parallelism on Distributed Memory Multiprocessor", IEEE Transactions on Parallel & Distributed Systems. To appear.
6. S.S. Pande. D.P. Agrawal, and J. Mauney, "A New Threshold Scheduling Strategy for SISAL Programs on Private Memory Machines." *Journal of Parallel and Distributed Processing*. vol. 21, no. 2. May 1994. pp. 223-236.
7. S. Darbha and D.P. Agrawal, "SDBS: A Task Duplication Based Optimal Scheduling Algorithm", Proceedings of Scalable High Performance Computing Conference. Knoxville, May 23-25 1994. pp. 756-763.
8. S. Darbha and D.P. Agrawal. "A Task Duplication Based Optimal Scheduling Algorithm For Varying Execution Time Tasks", Proceedings of 23rd International Conference on Parallel Processing. St. Charles. IL. August 15-19 1994. pp. 52-56.
9. Y. Potlapalli and D.P. Agrawal. "Conflict Analysis of Multistage Interconnection Networks". Proceedings of MASCOTS 1994. Jan 31- Feb. 2 1994. pp. 126-130.
10. H. Park, and D.P. Agrawal. "An efficient routing scheme for scalable hierarchical networks." Proc. 5th IEEE Symposium on Parallel and Distributed Processing. Dallas. Tx. Dec. 1-4. 1993, pp. 158-165.

11. Sandeep Kumar and Dharma P. Agrawal, "A Class Based Framework For Re-use Of Synchronization Code In Concurrent Object-Oriented Languages." *International Journal of Computers and Their Applications*. To Appear.

12. Sandeep Kumar, and D.P. Agrawal, "CORE: A solution to inheritance anomaly in concurrent object-oriented languages," Proc. International Conference on Parallel and Distributed Computing Systems, Louisville, KY, Oct. 14-16, 1993, pp. 75-81.

13. Y. Potlapalli, and D.P. Agrawal, "A new method for Hierarchical Interconnection of Processors," Proc. 22nd Int. Conference on Parallel Processing, Volume I Architecture, Aug. 16-20, 1993, pp. I 303-306.

14. T.Y. Chung, and D.P. Agrawal, "Design and Analysis of Multidimensional Manhattan Street Networks," *IEEE Transactions on Communications*, Vol. 41, No. 2, February 1993, pp. 295-298.

15. N. Sharma and D.P. Agrawal, "Hierarchical Distributed System Network Design with Cost-performance tradeoffs," Proc. Second International Symposium on High Performance Distributed Computing, July 20-23, 1993, pp. 336-343.

16. J.M. Conrad, and D.P.Agrawal, "Simulation of Generic Multiprocessor Configurations for Asynchronous Algorithms," *International Journal on Computer Simulation*, special issue on Multiprocessor Networks, Vol. 3, 1993, pp. 147-164.

17. S.S. Pande, D.P. Agrawal, and J. Mauney, "A fully automatic Compiler for Private Memory Machines," Proc. 26th Hawaii International Conference on System Sciences, Vol. II Software, Jan 5-8, 1993, pp. 263-274.

18. S.S. Pande, D.P. Agrawal, and J. Mauney, "Mapping Functional Parallelism in Distributed Memory Machines," Proc. The Second Sisal Users Conference, San Diego, Oct. 4-6, 1992, pp. 139-159.

19. J.M. Conrad, and D.P. Agrawal, "A Graph Partitioning-Based Load Balancing Strategy for a Distributed Memory Machine," 1991 International Conference on Parallel Processing, Vol. II Software, Aug. 17-21, 1992, pp. II-74-81.

20. J.M. Conrad, and D.P. Agrawal, "Distributed, Scalable, and Static Parallel Arc Consistency Algorithms on Private Memory Machines," Proc. Int. Conf. Distributed Computing Systems, June 9-12, 1992, pp. 442-449.

21. J.M. Conrad, D.P. Agrawal, and D.R. Bahler, "Scalable Parallel Arc Consistency Algorithms for Shared Memory Computers," Proc. 6th Int. Parallel Processing Symposium, March 23-26, 1992, pp.242-249.

22. S.S.Pande, D.P. Agrawal, and J. Mauney, "Palm: An Integrated Parallelism Enhancement Environment with Static-Dynamic Scheduling," Proc. of the Hawaii International Conference on System Sciences, Vol. II, Jan. 7-10, 1992, pp. 263-27.

23. N. Sharma, and D.P. Agrawal,"Hierarchical Reliability Evaluation of Large Networks," Proc. 3rd IEEE Symp. Parallel and Distributed Processing, Dec. 2-5, 1991, pp. 444-451.

24. J.M. Conrad and D.P. Agrawal, "Performance of an Asynchronous Parallel Algorithm on a Generic Multiprocessor Simulator," Proc. Pittsburgh Conference on Simulation and Modeling, PA, Part 3, May 1991, pp. 1290-1297.

25. S. Kim, S.S.Pande, D.P. Agrawal, and J. Mauney, "A Message Segmentation Technique to minimize task completion time," Proc. Fifth International Parallel Processing Symposium, April 30- May 2, 1991, pp. 519-524.

26. N. M. Kini, A. Kumar, and D. P. Agrawal, "Quantitative Reliability Analysis of Redundant Multistage Interconnection Networks," *American Mathematical Society/ACM DIMACS Volume 5* entitled "Reliability of Computer and Communication Networks," 1991, pp. 153-170.

27. S. S. Pande, D. P. Agrawal, and J. Mauney, "Impact of Control flow on processor allocation and utilization," (invited paper), *Frontiers in Parallel Computing*, Bhatkar et al. editors, Narosa Publishing House, 1991, pp. 163-175.

28. S. S. Pande, D. P. Agrawal, and J. Mauney, "Impact of Control flow on processor allocation and utilization," (invited paper), Proc. PARCOM90 Conference, Dec. 10-11, 1990, pp. 163-175.

29. S. S. Pande, D. P. Agrawal, and J. Mauney, "On Control flow and pseudo-static dynamic allocation strategy," Proc. 1990 Int. Conf. on Parallel Processing Vol. II Software, Aug 13-17, 1990, pp. 300-301.

30. D. P. Agrawal, W. E. Alexander, E. F. Gehringer, J. Mauney, and T. K. Miller, "B-HIVE: Hardware and Software for an Experimental Multiprocessor," Proc. 23rd Annual Hawaii International Conference on System Sciences, Vol. I, (Architecture Track), Jan. 2-5, 1990, pp. 40-47.

## 5 Scientific Personnel Supported by this Project

### 5.1 Graduated Students

1. Santosh S. Pande, Ph. D. Degree.
2. Chienhua Chen, Ph.D. Degree. (Summer support only)
3. James Michael Conrad, Ph.D. Degree. (Summer support only)
4. Viswanath Ramakrishnan, M.S. Degree. (Summer support only)
5. Nita Kini Sharma, Ph.D Degree. (Summer support only)

### 5.2 Current Students

1. Sekhar Darbha, Ph.D Student.
2. Sandeep Kumar, Ph.D Student.
3. Hyumin Park, Ph.D Student. (Summer support only)
4. Yashovardhan Potlapalli, Ph.D Student. (Summer support only)